Avoiding Pitfalls in Neural Network Research

G. Peter Zhang

***Abstract*—Artificial neural networks (ANNs) have gained extensive popularity in recent years. Research activities are considerable, and the literature is growing. Yet, there is a large amount of concern on the appropriate use of neural networks in published research. The purposes of this paper are to: 1) point out common pitfalls and misuses in the neural network research; 2) draw attention to relevant literature on important issues; and 3) suggest possible remedies and guidelines for practical applications. The main message we aim to deliver is that great care must be taken in using ANNs for research and data analysis.**

***Index Terms*—Data, model building, model evaluation, neural networks, pitfalls, publication bias, software.**

I. Introduction

ARTIFICIAL neural networks (ANNs) have enjoyed considerable popularity in recent years. They have been used increasingly as a promising modeling tool in almost all areas of human activities where quantitative approaches can be used to help decision making. Research efforts in ANNs are considerable, and the literature is vast and growing. This trend will continue in the foreseeable future. Indeed, ANNs have already been treated as a standard nonlinear alternative to traditional models for pattern classification, time series analysis, and regression problems. In addition to numerous standalone software devoted to neural networks, many influential statistical, machine learning, and data-mining packages include neural network models as add-on modules in their recent editions.

The popularity of ANNs is, to a large extent, due to their powerful modeling capability for pattern recognition, object classification, and future prediction without many unrealistic *a priori* assumptions about the specific model structure and data-generating process. The modeling process is highly adaptive, and the model is largely determined by the characteristics or patterns the network learned from the data in the learning process. This data-driven approach is highly applicable for any real-world situation where theory on the underlying relationships is scarce or difficult to prescribe but data are plentiful or easy to collect. In addition, the mathematical property of the neural network in accurately approximating or representing various complex relationships has been established and supported by solid theoretical work [9], [15], [23], [34], [48], [59], [64]–[67], [125], [126]. This universal approximation capability is important because many decision support problems such as pattern recognition, classification, and forecasting can be treated as function mapping or approximation problems.

Despite the growing success of neural networks, we have seen many problems, pitfalls, and misuses frequently emerge with neural network research and applications in the literature. ANNs' powerful pattern recognition ability and flexible modeling approach make them attractive, bringing with them great opportunities and the strong potential to be useful in solving real-world problems. But at the same time, this tremendous flexibility and wide applicability also exposes them to the real danger of inappropriate uses. For example, ANNs have recently been promoted as a data-mining tool to search for valuable information in a large database. However, it is often too easy for the technique to be used instead for data dredging or data snooping [20], [127]. The real danger here is that even if there is no useful information in the data, neural networks may still find something "significant," misleading unwary users. As Salzberg [104] points out, "when one repeatedly searches a large database with powerful algorithms, it is all too easy to 'find' a phenomenon or pattern that looks impressive, even when there is nothing to discover." This danger of data snooping is also discussed by many researchers including [22], [58], and [127].

Pitfalls can arise in the use of many quantitative methods. This can happen when researchers do not have a complete understanding of the technique or a careful design to avoid possible abuses, especially when the technique can be "easily" implemented with an automatic software package. It is well known that statistics is often misused [2], [62], [69], [75]. As early as in 1938, Cohen [29] observed various pitfalls in the use of descriptive statistics in practice. King [76] identifies a set of serious statistical mistakes appearing in the quantitative political science literature. Chatfield [17] gives many examples of common modern-day pitfalls in statistical investigations and comments that "statistics is perhaps more open to misuse than any other subject, particularly by the nonspecialist." Misuses of discriminant analysis are detailed in [39] for business, finance, and economics applications, in [147] for medical diagnoses, and in [146] for psychology problems. The abuse of variance models in regression is discussed in [118]. Issues of statistical pitfalls related to model uncertainty and data dredging are discussed in [20] and [21].

The fundamental principle of ANNs for data analysis and modeling is the same as or similar to that of statistics, and in many aspects ANNs can be treated as the nonlinear counterparts of statistical techniques [24], [28], [58], [101], [102], [108], [111], [124]. Thus, pitfalls in statistical analysis are likely seen also in neural network research. For example, Schwartzer *et al.* [140] list six major types of ANN misuses, which are similar to those observed in [147] with statistical discriminant analysis. While any quantitative method is subject to misuses, methods that are complicated, automatic, and new are generally more likely to be misused than simple, nonautomatic, and established methods. Because of their newness, complexity, and lack

Manuscript received February 5, 2003; revised August 3, 2005. This paper was recommended by Associate Editor Jun Wang.

The author is with the J. Mack Robinson College of Business, Georgia State University, Atlanta, GA 30303 USA (e-mail: gpzhang@gsu.edu).

Digital Object Identifier 10.1109/TSMCC.2006.876059

of standard modeling procedure, as well as the "black-box" nature, ANNs are even more susceptible to the danger of misuses than other subjects including statistics. These probably are the reasons that we see more controversial or contradictory results reported in neural network research.

Pitfalls and abuses in the neural network research are harmful to the field. Indeed, skeptical opinions regarding neural networks as hype or passing fad are abundant [18], [19], [55], [129]. Some of this skepticism may be justifiable given a large number of problems observed in the neural network community. For example, Schwarzer *et al.* [140] review the literature of neural network applications for prognostic and diagnostic classification in oenological studies between 1991 and 1995 and find the following seven misuses: 1) mistakes in estimation of misclassification probabilities; 2) fitting of implausible functions; 3) incorrectly describing the complexity of a network; 4) no information on complexity of the network; 5) use of inadequate statistical competitors; 6) insufficient comparison with statistical method; and 7) naïve application to survival data. They conclude that "there is no evidence so far that application of ANNs represents real progress in the field of diagnosis and prognosis in oncology."

Therefore, in order for the field to grow in a healthy direction and achieve significant advances in the future, it is important for researchers to be aware of potential pitfalls as well as ways to avoid them. The goals of this paper are to: 1) point out various common pitfalls and misuses in the neural network research; 2) draw attention to relevant literature on important issues; and 3) suggest possible remedies and guidelines for practical applications. We will mainly focus on the multilayer feedforward type of neural networks, although many issues discussed could also be applied and extended to other types of neural networks. The focus on feedforward neural networks is due to their popularity in research and applications as according to Wong *et al.* [151], about 95% of business applications of neural networks reported in the literature use this type of neural model. The main message we aim to deliver is that great care must be taken in using ANNs for research and data modeling.

The remainder of the paper is organized as follows. Section II provides several major factors that contribute to the common pitfalls in neural network applications. Sections III–IX discuss various pitfalls in neural network research as well as recommended approaches to avoid them. Finally, Section X provides concluding remarks.

II. Factors Causing Common Pitfalls

Pitfalls in neural network research arise in many different forms due to various factors. The most important contribution to the many pitfalls is perhaps the nonlinear nonparametric nature of the neural network model. While this property is desirable for many real-world applications, it also brings about more opportunities to go wrong in the modeling and application process. Compared to their linear statistical counterpart, neural networks have fewer assumptions, more parameters to estimate, many more options to select in the modeling process, all of which open more possibilities for inappropriate uses and problematic applications. As Granger [56] puts it, "building models of nonlinear relationships are inherently more difficult than linear ones. There are more possibilities, many more parameters and thus more mistakes can be made."

The second major reason is the lack of a uniform standard in building neural network models. For example, numerous nonlinear algorithms that are alternatives or variations to the basic backpropagation (BP) algorithm exist. These algorithms vary in efficiency and effectiveness in estimating parameters. In addition, different and sometimes conflicting guidelines are provided on many factors that could affect ANN performance. The problem is that ANN models are sensitive to many of these factors.

Pitfalls are more likely to occur to unwary researchers who lack the expertise and knowledge of the various forms of abuses. They often have the inappropriate supposition that ANNs can be built with automatic software, and that users do not need to know much of the model detail.

Another reason that many inappropriate uses of ANNs are published is the lack of details on several key aspects of the model-building process. Authors or researchers often do not give sufficient detail, essential features, or adequate description of their study methodology, which hinders easy understanding or replications for others. On the other hand, reviewers may not pay attention to these issues. The lack of transparency, thus, contributes to errors in published research work.

III. Black Box Treatment of ANNs

Neural networks are often treated and used as black boxes. In a survey by Vellido *et al.* [119], the lack of explanatory capability in terms of the "incapacity to identify the relevance of independent variables and to generate a set of rules to express the operation of the model" is considered by researchers as the main shortcoming in the application of neural networks. While it is true that ANNs are not able to give the same level of insight and interpretability as many statistical models, it is a pitfall to treat them as complete black boxes with the assumption that we know nothing about the nature of the ANN model built for a particular application except for the output estimate or prediction. Often, "black box" is used either as an excuse to relieve researchers from exploring further inquiries and examining the established model more rigorously or as a justification for automatic modeling so that people with little knowledge of neural networks and subject matter can do the modeling easily [63]. With this view, users do not need to understand how the model works and formal statistical tests may not be applied to test the significance of the model and the parameters. Faraway and Chatfield [43] have pointed out the potential danger of the opinion that ANN models can be built blindly in "black box" mode.

Advances in ANN research have suggested that neural networks are not totally unexplainable. In fact, there are considerable research interests in offering insights into the "black box" operation of neural networks. One active research area is in the understanding of the effect of explanatory variables on the dependent variable or output of the model. Numerous measures have been proposed to estimate the relative importance or contribution of input variables. Some of these measures are reviewed in [130]. Intrator and Intrator [71] propose a method based on the robustification technique to interpret neural network results in terms of the input effects and interactions among input variables. Another area of research is in rule or knowledge extraction from trained networks. Benitez *et al.* [10] and Castro *et al.* [16] establish the equality of ANNs to fuzzy-rule-based systems and propose the methods to translate the knowledge embedded in the neural networks into more understandable fuzzy rules. Setiono *et al.* [135], [136] propose the algorithms to discover rules from networks for regression problems. Andrews *et al.* [4] and Tickle *et al.* [116] survey the techniques used for extracting rules and knowledge embedded in trained ANNs. Tickle *et al.* [117] even conclude that after more than 10 years of research in the knowledge discovery in ANNs, we have already reached a point where the deficiency of the black box nature is "all but redressed."

It is a well-known fact that, for classification problems, ANNs provide direct estimates of the posterior probabilities [52], [100], [122]. The interpretation of neural network outputs as posterior probabilities is of fundamental importance because many traditional Bayesian classification methods are established on the ability to estimate the posterior probability. As summarized in [122], "Interpretation of network outputs as Bayesian probabilities allows outputs from multiple networks to be combined for higher level decision making, simplifies creation of rejection thresholds, makes it possible to compensate for difference between pattern class probabilities in training and test data, allows output to be used to minimize alternative risk functions, and suggests alternative measures of network performance."

The links and equivalence between ANNs and various traditional statistical models have been well established. For example, Raudys [97] shows that decision boundaries of single-layer perceptrons are equivalent or close to those of the seven statistical classifiers. Gallinari *et al.* [49] and Schumacher *et al.* [105] establish the theoretical connection of ANNs to discriminant analysis and logistic regression. Certain ANN models have been suggested to be equivalent to conventional time series models. For example, autoregressive models can be implemented via neural networks [31]. Connor *et al.* [30] demonstrate that recurrent neural networks are a special case of nonlinear autoregressive and moving average models, while feedforwared ANNs are a special case of nonlinear autogressive models. Therefore, the fundamental mechanism of neural networks is the same as or very similar to many statistical methods in classification and forecasting.

Many people believe that the functional form of the neural network model cannot be known precisely. In addition, the exact relationship between inputs and outputs is too complex to express. This is not true. The general functional form of a single hidden layer feedforward neural network can be written as follows:

$$y_k = \alpha_{0k} + \sum_{j=1}^{q} \alpha_{jk} f\left(\sum_{i=1}^{p} \beta_{ij} x_i + \beta_{0j}\right) + \varepsilon, \quad k = 1, 2, \ldots, r \qquad (1)$$

where $\{x_i\}$ and $\{y_k\}$ are the vectors of the input and output variables, respectively; p, q, and r are the numbers of input, hidden, and output nodes; f is a transfer function such as the popular logistic $f(x) = 1/(1 + \exp(-x))$; $\{\alpha_{jk}\}$ and $\{\beta_{ij}\}$ are the sets of weights from the hidden to output nodes and from the input to hidden nodes, respectively; and α_{0k} and β_{0j} are weights of arcs leading from the bias terms, which have values always equal to 1. Equation (1) suggests that the neural network model is a weighted sum of a basis function of a linear combination of input variables.

Of course, the model parameters such as p, q, and r, as well as $\{\alpha_{jk}\}$ and $\{\beta_{ij}\}$, will vary from application to application. But the basic relationship specified in (1) is always held. Thus, it is not difficult to express the trained neural network model with an exact mathematical relationship, although it can be complex. This knowledge along with the estimates of weights is often useful as one may wish to perform further analysis to explore the property of the relationship and extract the knowledge embedded in the connecting weights. Therefore, totally ignoring the internal workings of ANNs could result in inappropriate treatment of neural networks and loss of the opportunity to gain insights from the established model.

IV. Overfitting and Underfitting

Overfitting is one of the most cited problems with ANNs. The topic is well discussed and every neural network researcher is perhaps aware of the danger of overfitting. Overfitting limits the generalization ability of predictive models. For neural networks, it is easy to get a good or excellent result on the in-sample data, but this by no means suggests that a good model is found. It is likely that the model memorizes noises or captures spurious structures, which will cause very poor performance in the out-of-sample data. Overfitting typically happens when users build overly large neural networks and/or the in-sample data used to train networks are small. Therefore, it is more likely to occur with ANN models than most statistical models due to their flexible modeling approach and the large number of parameters to be estimated from the data.

The above fact is well known. The guidelines and techniques to avoid overfitting are plentiful. Unfortunately, the dangers of overfitting "are not always heeded" [21]. In the literature, we see many applications with inappropriate model sizes relative to sample sizes. For example, Fletcher and Goss [44] use 36 observations in their study of bankruptcy prediction application. Davis *et al.* [138] have only 32 observations in training a neural network with more than 200 input nodes. Adya and Collopy [1] find that, among 27 effectively validated studies in forecasting and prediction, only three attempt to control the potential problem of overfitting. In a review for auditing and risk assessment applications, Calderon and Cheh [137] report that most studies suffer from the overfitting or overtraining problems. Business applications listed in [79] and [119] also indicate many overfitting problems. In a survey on 43 applications of neural networks for the forecasting of water resources variables, Maier and Dandy [84] find that, for most applications, the relationship between the network size in terms of the connection weights and training sample size is ignored and the number of hidden nodes used is greater than the theoretical upper bounds.

Hippert *et al.* [61] review 40 papers in the applications of ANNs for short-term load forecasting and conclude that "most of the papers proposed ANN architectures that seemed to be too large for the data samples they intended to model.... These ANNs apparently overfitted their data."

Another related pitfall is to include as many input variables as possible in the model, believing or hoping that the ANN can identify the most important and relevant variables through the linking weights' adjustment during the model-building process. Including a large number of unnecessary variables not only increases the model complexity and the likelihood of overfitting, but also causes more time and effort wasted in training. Moreover, the true pattern may be masked by the irrelevant factors and their interactions. In [138], for an application of neural networks for audit control risk assessment, 210 input variables are used with only 32 observations in the training sample.

On the other hand, underfitting occurs if a neural network model is under-specified or not trained well. With underfitting, the model does not give good fit even to the training set. While underfitting is usually not a major concern compared to overfitting, ignoring the underfitting can also cause problems in applications, especially when the training algorithm is not appropriately used to guarantee a good solution in the estimation process.

The phenomena of under- and overfitting are well known in the statistical literature and are well discussed in the neural network community with the concept of bias/variance tradeoff [50]. A large number of research publications have appeared on the bias/variance issues of neural networks learning (see [130] for relevant research activities in the classification area). Although most studies in the area are theoretical in nature, it is helpful for ANN users to have a clear understanding of the basic issue of bias versus variance.

Because of the over- and underfitting problems, it is often desirable to report the training process and both the in-sample and out-of-sample performances. Published research studies, however, do not do well in reporting the in-sample results. Although the main focus and interest of ANN applications are on the out-of-sample performance, it may be difficult to comprehensively judge the quality of the model without also looking at the in-sample results. Adya and Collopy [1] find that the most common problem for effective implementation of the model in published studies is their failure to report in-sample results. They further advise, "if a study does not report in-sample performance on the network, we suggest caution in acceptance of its *ex ante* results."

Building a parsimonious model with minimum number of input variables and parameters but at the same time achieving high predictive accuracy is critical to avoiding under- and overfitting problems. To this end, researchers should combine both qualitative domain knowledge and quantitative variable selection approach to select the most important predictor variables. Additionally, node pruning and weight elimination methods can be used to reduce the complexity of the neural model [99]. For recent reviews on statistics-based and neural-network-based feature variable selection techniques, readers are referred to [72] and [130].

V. Data-Related Problems

ANNs are data-driven methods. Without data, it would be impossible to build an ANN model in the first place. However, what data should be used and what quality characteristics the data should possess are rarely considered by ANN researchers. In many applications, data are used as if they are free of errors and are representative of the true underlying process. This is certainly not necessarily true in many situations. It is well known that neural networks as well as other modeling techniques cannot turn garbage inputs into golden information, or "garbage in and garbage out." Consequently, the reliability of neural network models depends to a great extent on the quality of data.

Data used in neural network research typically come from two sources: The primary source and the secondary source. The primary data source is the original data collected with a specific research question to be investigated. On the other hand, the secondary data source contains data sets that have previously been used for other purposes. Both primary and secondary data have been used in neural network research, although the secondary data source is used much more frequently due to its convenience and much less effort to obtain. Secondary data are often used for the purposes of model evaluation, model comparison, and methodology illustration.

Data stored in organizational databases often contain significant errors [80], which can affect the predictive accuracy of neural network models. While Bansal and Kauffman [6] find that ANN models are more robust than linear regression models when data quality decreases, results reported in [78] suggest that error rate and its magnitude can have substantial impact on neural network performance. Klein and Rossin [78] believe that an understanding of errors in a data set should be an important consideration to ANN users and efforts to lower error rates are well deserved.

In addition to errors in the data set, data representativeness is another issue that is largely ignored in the literature due to the convenience of sample selection. If the data are not random or not representative of the true population, results obtained from the neural network analysis may not be useful or generalizable. Few studies, however, examine this issue explicitly. In conventional time series forecasting, it is well known that nonstationarity can have significant impact on the analysis and forecasting and preprocessing are often necessary to make data stationary. In the neural network literature, most studies do not consider the possible effect of nonstationarity. Although some researchers explicitly address the issues of model uncertainty and the shift of underlying data-generating process, others overlook them entirely, even if the data may indicate some potential problems.

An interesting exception to the above representativeness issue is in a two-group classification with ANNs. Although group composition of the two classes in the population is rarely equal (and in many cases extremely unbalanced), it is often beneficial to include equal number of examples from both the classes to ensure a good representation of the small group [11], [131]. Wilson and Sharda [128] and Jain and Nag [73] study the effect of training sample composition on neural classifier performance and find that using balanced samples from two groups to build the network model can yield the best prediction results on the holdout sample, even if the holdout sample is representative of unbalanced population. Thus, using unrepresentative in-sample data in this context is warranted. In other words, using a "representative" sample can result in a suboptimal solution in the practical application of the neural classifier.

Many studies have used data sets stored in online repositories such as the University of California, Irvine (UCI) machine learning repository and several well-known forecasting competition databases. Often these public data sets are used to either demonstrate the proposed new methodology or compare different methods. It is advantageous to have a common convenient database to serve as a benchmark to test new methods. Fisher's iris data for classification and Wolf's sunspot series for nonlinear time series forecasting are two well-known data sets used by numerous researchers for many decades. But the problem with using the same data set repeatedly is that significant results may be "mere accidents of chance" [104]. Individual data snooping is possible if an individual becomes familiar with the characteristics of some data sets and develops specific algorithms tailored to these data sets. Denton [36] and Salzberg [104] analyze the danger of the so-called "collective data mining" and show that using the same data set by more than one investigator distorts the Type I error rate—the probability of making at least one mistake. The publication bias against nonsignificant results can further exacerbate the problem [8].

Another problem with the use of public data is that the data set may not be a random or unbiased sample even though the data repository contains a large number of data sets. For example, the well-known M1 and M3 forecasting competition data repositories comprise more than 1000 and 3000 business and economic time series, respectively, with different types and periodicities. Yet, few agree that they are random or representative samples of all possible data series [85], [90]. Therefore, it is a pitfall to try to generalize the results beyond the data sets tested. One possible solution to this problem is to increase the data repository over time to improve representativeness and to be careful in interpreting results obtained from using such data sets. Another solution is to use artificial or simulation data to control the properties of the data for which new methods or algorithms are targeted. Thus, the use of artificial data can "test more precisely the strengths and weaknesses" [104] of a new method.

Sample size of the data set is another important issue in all quantitative modeling endeavors including neural network analysis. Some techniques have a higher requirement on minimum size for data modeling and analysis. For example, in the time series forecasting context, Box and Jenkins [12] suggested that at least 50, or better 100, observations are necessary to build a successful autoregressive integrated moving average (ARIMA) model. Neural networks are nonlinear nonparametric models that typically require larger sample sizes than conventional statistical procedures for model building and validation. In general, the larger the sample is, the better is the chance for a neural network to adequately approximate the underlying complex patterns without suffering from the problem of overfitting. Raudys and Jain [98] study small sample size effects and find that small sample size can make the problem of designing a pattern classifier very difficult. On the other hand, we have seen reports that larger sample sizes do not always result in better out-of-sample performances [120]. This could also be true for time series forecasting problems in which data may not be stationary or may contain structural changes over time.

The literature certainly does not give specific guidance on the sample size requirement for particular applications other than the general recommendation for larger samples. Indeed, there is no such thing as "one size fits all" because the appropriate sample size depends on many factors such as the complexity of the problem, the number of input variables, the number of parameters in the model, and the noise in the data. Neural network researchers, therefore, must be cognizant of the sample size issue in designing their particular models. In particular, smaller sample size requires researchers to pay closer attention in selecting model parameters. Though sample size is often constrained by the availability of data, the practice of simply accepting a data set regardless of its sample size should be avoided. If the sample size is too small, remedial actions such as resampling or cross-validation techniques may not be very helpful.

An important issue related to sample size is data splitting, or dividing the data into two portions: an in-sample and an out-of-sample. The in-sample is used for model fitting and estimation while the out-of-sample or holdout sample is used to evaluate the predictive ability of the model. Because of the bias/variance concern, it is critical to test the ANN model with an independent holdout data set, which is not used in neural network training. That is, we must set a portion of the whole available data aside and never touch it in the model-building process. This practice is necessary to ensure that the model finally built has a true value for practical uses. As a consequence, the sample size used to train the network is smaller than the total number of available data points. The true size for model building may be further reduced if the in-sample data are further divided into a training set for model estimation and a validation set for model selection.

In a recent survey of 43 papers in forecasting water resources variables using ANNs, it was found that "data division was carried out incorrectly in most papers.... The proportion of data used for training and validation varied greatly. Generally, the division of data was carried out on an arbitrary basis and the statistical properties of the respective data sets were seldom considered" [84].

There is no consensus on how to divide the data into an in-sample for learning and an out-of-sample for testing. Picard and Berk [142] suggest that 25%–50% data are used for validation for linear regression problems and if the emphasis is on parameter estimation, fewer observations should be reserved for validation. According to [22], forecasting analysts typically retain about 10% of the data as holdout sample. Granger [56] recommends that, for nonlinear modeling, at least 20% of any sample should be held back for an out-of-sample evaluation. Michie *et al.* [88] also recommend holding back approximately 20% of the data for testing and dividing the remaining data into a training set and a validation set. Hoptroff [63] suggests using 10%–25% of data as the testing sample while Church and Curram [27] use "more conservative 30%." Many other different splitting strategies have been used in the literature. When the in-sample data need to be further split into a training sample and a validation sample, the issue is more complicated. Hastie *et al.* [60], however, note a typical division of 50% for training and 25% each for validation and testing.

It is important to note that the issue of data splitting is not about what proportion of data should be put in each subsample. But rather it is about an adequate sample size in each sample to ensure sufficient learning, validation, and testing. As the available data size varies dramatically from application to application, the number of observations used in each sample can differ greatly with the same proportion of data for testing purposes. Although statistical methods such as regression can have as much as 50% of the data used for testing [109], most neural network applications may not be able to afford that large portion and typically a much smaller percentage such as 10% or 20% should be considered. Nevertheless, it is important to include a sufficient number of observations in the test sample so that the model's generalization ability can be evaluated adequately. Hoptroff [63] recommended at least 10 data points in the test sample, while the study by Ashley [5] suggested that much larger out-of-sample size is necessary to achieve statistically significant improvements for forecasting problems. Although more test data are desirable to ensure that the test sample performance is not due to chance, the tradeoff has to be made between the in-sample size and the test sample size and typically more data should be allocated for ANN model construction.

The lack of guidelines does not mean that data splitting may be done arbitrarily. When the size of available data set is large (e.g., more than 1000 observations), different splitting strategies may not have a major impact on adequate learning and evaluation. But it is quite different when the sample size is small. In addition, splitting generally should be done randomly to make sure each subsample is representative of the population. There is no question of doing this for regression and classification problems. On the other hand, time series data are difficult or impossible to be split randomly. For time series problems, data splitting is typically done at researchers' discretion. However, it is still important to make sure that each portion of the sample is characteristic of the true data-generating process. LeBaron and Weigend [81] evaluate the effect of data splitting on time series forecasting and find that data splitting can cause more sample variation, which in turn causes the variability of forecast performance. They caution the pitfall of ignoring variability across the splits and drawing too strong conclusions from such splits. Their finding is in line with that of Faraway [41] who shows that for regression modeling, data splitting may increase variability in estimates. Furthermore, data splitting may lose efficiency and effectiveness in different contexts (see [21], [32], and [42]).

VI. Model Building

Building a successful predictive neural network model is not an easy task. There are many possible ways to build an ANN model and a large number of choices to make during the model-building process. Numerous parameters and issues need to be considered and experimented with before a satisfactory model may emerge. Adding to the difficulty is the lack of standards in the process and there are a great number of controversial rules of thumb and guidelines in the literature. It is important to note that most empirical rules work only for special problems or situations, and therefore, treating these rules as universal and using them blindly is a pitfall that should be avoided.

The major decisions a designer or a builder of a neural network model must make include data preparation, input variable selection, the network architecture parameters such as the number of input, hidden and output nodes, node connection, training algorithm, transfer functions, and many others. Some of these issues must be solved before actual model building starts while others are determined during the model-building process. Neural network design should be treated as a more important issue than the subsequent analysis because if there are flaws in the design of ANN model building, then further analyses are worthless no matter how good they may look. Unfortunately, great emphasis is often placed on the analysis of the results rather than on good design issues in neural network research.

Data preprocessing is often recommended and used to highlight important relationships and create more uniform data to facilitate network learning, meet algorithm requirements, or avoid computation problems. However, the necessity and effect of data preprocessing on neural network learning and prediction is still undecided as research findings are often contradictory. Some researchers conclude that because of the universal approximation capability, data preprocessing is not necessary and the model can pick up all the underlying structure from the raw data. For example, Gorr [54] believes that neural networks should be able to simultaneously detect both the nonlinear trend and the seasonality in the data. Earlier studies on seasonal time series forecasting found that neural networks are able to directly model the seasonal behavior and preseasonal adjustment is not necessary [107]. Recent studies, however, suggest that predeseasonalizing data is critical in improving forecasting performance [89], [145]. Callen *et al.* [14] report discouraging results with neural networks for predicting quarterly accounting earnings. One of the potential reasons is that they do not consider data preprocessing such as deseasonalization. Thus, ignoring the potential effect of data transformation or using inappropriate data preprocessing techniques can reach quite different results or conclusions.

One related problem with data transformation is that results with transformed data will have different scale than the original data. To better interpret the results or to compare them with other methods built with raw data, the outputs from the ANN models need to be scaled back to the original data range. From a practical point of view, the accuracy measure obtained by the ANNs should be based on the original data scale. In many studies, however, researchers fail to indicate whether the performance measures are calculated on the original or transformed scale.

Determining appropriate neural network architectures is one of the most important tasks and numerous guidelines are available. Yet, many pitfalls have been observed in building and selecting ANN models.

For most regression and classification problems, the numbers of input and output nodes are usually determined based on prior or subject matter knowledge. For the time series forecasting problems, however, both numbers need to be determined via experimentations. In particular, the choice of input variables is critical [133]. One of the pitfalls is the use of the linear method to select model parameters. For example, principle component analysis (PCA) has been used for feature selection in regression and classification problems and the ARIMA model has been suggested and employed for selecting input lag structure in time series forecasting [83], [114]. Park *et al.* [91] use both linear AR model to identify the input lag structure and PCA to determine the number of hidden nodes. Balkin and Ord [141] apply the stepwise regression approach to select the inputs to neural networks. The inappropriateness of these methods is that they cannot capture nonlinear structures. In addition, linear models such as PCA are unsupervised learning procedures and do not consider the correlation between dependent variables and input variables.

Another related problem in determining input variables is the tendency to throw a large number of variables to the model regardless of their relevance or redundancy, hoping ANNs can pick the most appropriate input variables by adjusting the linking weights. The potential effects of this practice are the overfitting, masked patterns, and increased modeling time. On the other hand, choosing input variables "too carefully" via data snooping and then reporting the best results as if the input variables are chosen in the first place can be even more dangerous.

Almost all studies in time series forecasting use one output node for both one-step forecasting and multistep forecasting. While single output node networks are suitable for one-step forecasting, they may not be effective for multistep forecasting situations as empirical findings [87] suggest that a forecasting model best for a short term is not necessarily good for a long term. For this and other reasons [132], it is recommended that multiple output nodes be used for multistep forecasting situations. This is consistent with the suggestion in [21] and [22] to use different models for different lead times.

Neural networks with single hidden layer have been shown to have universal approximation ability and they are also relatively easier to train. This is the reason that two or more hidden-layered networks are rarely used in applications. However, excluding more hidden layers from considerations may cause inefficiency and poor performance in neural network training and prediction, especially when a one-layer model requires a large number of hidden nodes to give desirable performance. Research studies in [77], [110], and [134] show that two-hidden-layer networks can provide more benefits for some problems.

The number of hidden nodes determines not only the network complexity to model nonlinear and interactive behavior but also the ability of neural networks to learn and generalize. Too many or too few will cause the overfitting or underfitting problem. Unfortunately, there is no unique magic formula that can be used to calculate this parameter before training starts and it usually must be determined by the trial and error method. Interestingly, if the number of hidden nodes could be predetermined, ANNs would not be called a "data-driven" method because hidden nodes to a large extent determine the neural network model. Although empirical formulas or rules are plentiful, users should be careful in applying them. In the literature, some studies have blindly used previous empirical rules without further exploring the possibility that the number is not optimal for their particular applications, while others choose a particular amount without reporting how it is obtained.

Training a neural network is a complicated issue because of the nonlinear optimization involved. A good training algorithm can make a difference in adequate model estimation. Therefore, users should use more efficient algorithms whenever possible. Because of the local minima problem inherent in nonlinear optimization procedures, finding a global optimal or better local solution is the goal in the training process. Because of the sensitivity of neural network estimation to the initial conditions, using multiple random starting points to reduce the risk of bad local minima is often recommended. Nevertheless, many studies still use older less efficient BP algorithms due to easy access and availability in software, and do not consider multiple training methods. Curry and Morgan [33] discussed many problems with the basic BP training algorithm.

Common practice in building a neural network model is to divide the available data into two portions: An in-sample for model building and a holdout sample or out-of-sample for model testing or assessment. The in-sample data may be further split into a training sample for model fitting and a validation sample for model selection. It is important to note that except for the training sample, the nomenclature for the other two samples is not used consistently in the literature. That is, a "validation (or test) sample" in one study may become a "test (or validation) sample" in another one. It can be further complicated if all available data are divided into only two portions, the later part is sometimes called "test" sample while other times "validation." This inconsistency may cause confusion to some researchers.

The major pitfall in ANN model building is the use of the whole data set to do model estimation and model selection. This can happen in two forms. The first is that researchers divide the data into only two portions of a training set and a test (or validation) set and then choose the model based on the best performance of the test set. The second is that researchers do have three portions of training, validation, and test samples. But rather than using the last sample as an independent one for model evaluation, they use it repeatedly to fine tune the model estimation and selection process conducted on the first two parts of the data. Of course, without appropriate evaluation, the model developed may not have any value for practical uses as the model is tailored too much to the data on hand and the true performance on unseen data cannot be assessed. However, this lack of independent holdout samples for a genuine out-of-sample evaluation is fairly common in published research as Duin [38] points out, "there will be a strong temptation for the researcher to do some more tuning when he finds out that his neural network performs relatively badly. Papers that emphasize that this has not been done are very rare." Lisboa [148, p. 29] notes that in most medical applications of ANNs, "there is no attempt to separate a design data set, used for training and parameter tuning, or testing, from a validation set used for performance estimation."

It should be noted that if the cross-validation approach is used to select the best model, then the validation sample result should not be treated as the true performance of the model. To test a model selected this way, an independent validation sample must be used. It is, however, possible to combine the first two parts of the data (training and validation) to reestimate the model parameters. Of course, there is no guarantee that this will yield better results out of sample. If this strategy is used in the research, it is important to make it clear in the publication.

It is important to note that it is possible to use only two data sets for model building and testing. This is typically the case when researchers apply some special in-sample model-building and selection procedures. Some of the pruning methods such as node and weight pruning [99] as well as constructive methods such as the upstart and cascade correlation algorithms [40], [47] belong to these methods. In-sample model selection approaches based on traditional information-based criteria such as Akaike's information criterion (AIC) and Bayesian (BIC) or Schwarz information criterion (SIC) have been proposed and used in time series studies [21], [43]. However, these in-sample model selection criteria are developed based on the assumption of asymptotic normality of the maximum likelihood estimators. Although suitable and commonly used for linear parametric models, in-sample criteria are not directly applicable to and theoretically justified for neural networks [3]. Furthermore, the effectiveness of these criteria for nonlinear neural network models has not been supported by empirical studies as Swanson and White [112], [113] and Qi and Zhang [95] find that the in-sample criteria such as AIC and BIC cannot provide a reliable guide to out-of-sample prediction performance. The use of these information-based selection criteria in practice, therefore, should be with caution.

Another major pitfall in the published research is the lack of detail of the model-building process. Some studies simply list the architectures used without giving any indication on how a particular architecture is selected. Others give some vague justification without giving further detail. In fact, statements similar to "our networks used ten nodes in the hidden layer, which was found to be sufficient for all the models" [27] and "we tried and tested a number of different architectures of the neural network.... The results reported here are based upon the best ANN forecast" [14] are common. This lack of clear documentation of the model-building process is a serious problem reported in several recent surveys of ANN applications [61], [79], [84].

Without sufficient detail of the modeling process, it is difficult or impossible to judge if the research design is conducted appropriately as well as how much tuning is done. Of course, tuning the parameters in the training set is acceptable. But if the test set is also involved, then it is problematic. Furthermore, as repeated tuning of modeling parameters could be done during the process, it is impossible for others to replicate the study if the authors do not report this tuning process as results obtained from neural networks can vary dramatically depending on numerous factors including weight initialization, learning rate, momentum, training length, stopping condition, and whether or not using special techniques such as weight decay and node pruning are used. As replicability is a critical principle of scientific research, lack of detail can be detrimental to the neural network field.

One recent example of the importance of replication is given by Racine [96] who tries to repeat a previous study by Qi [94]. Although the complete design was not detailed in the original article and recalled from the author, Racine was able to conduct an approximate replication study by using the same data, software, and modeling approach as in Qi [94] as well as several different scenarios when exact detail was not available. His results suggest that "both replicability and the claimed superiority of the ANN are elusive" [96, p. 380]. Another example is the attempt by Zhao *et al.* [143] to replicate a previous study by Hill *et al.* [144] for a time series forecasting. They find that while the general conclusion of good neural network performance is still valid, they are not able to achieve the same magnitude of the improvement reported in [144] due to the lack of information on several key factors in the original study, which does not permit reproducing them.

Thus, it is imperative to report the detail of the ANN model design and model-building process. The minimum detail should include the architectures experimented, data splitting and preprocessing, training settings such as weight initialization method, learning rate, momentum, training length, stopping condition, algorithm used, and model selection criterion. If special procedures such as regularization, weight decay, or node pruning are employed, it is necessary to give the detail or references.

VII. Software Uses

There are many software packages available in ANNs, ranging from stand-alone freeware or shareware to expensive commercial packages. These packages vary greatly in features, options, training algorithm, programming capability, and user interface. While the availability of powerful and easy-to-use ANN software greatly enhances the research capability and research activities, it also increases the risk of misuse and error [33]. The real danger is the tendency for ANN users "to throw a problem blindly at a neural network in the hope that it will formulate an acceptable solution" [46].

It is unwise to believe each software package has the same capability to perform modeling and predicting tasks and can be relied upon to give satisfactory results. It is especially dangerous to believe that the software can be used in a purely automatic mode and that users without much knowledge of ANNs can build ANN models easily and successfully. If a user does not have knowledge of the many important issues in ANN model building mentioned earlier and are not aware of the choices, assumptions, default settings, training algorithm, and limitations of the package, it is not unlikely that dubious results will be generated. Even with a good solid software package, users still face a large number of important choices and decisions to make, and using the default settings is not always the best strategy.

Most ANN software packages serve only as a convenient means to do nonlinear optimization inherent in any neural network training. They do not address adequately the issue of data splitting, model selection, and model evaluation. The method used for choosing the number of parameters is often not well implemented. In addition, many vendors of ANN software do not address the overfitting problem and use the result from the training data as a selling point for their algorithms [35].

MATLAB Neural Network Toolbox is one of the most popular commercial packages in the market. Yet, many problems experienced with the software such as the reliability and numerical accuracy have been reported by Gencay and Selcuk [51].

Curry and Morgan [33] raise the concern of inappropriate use of training algorithms, especially the widely used BP algorithm in many ANN software packages. Because of its popularity, the BP algorithm may be the default of many packages even though it has many weaknesses. Because of this limitation and because of unwary users who may not be aware of the internal workings of the algorithm, "commercially available software should be used with great care" [33, p. 131].

Therefore, it is critical to fully understand the capabilities as well as the limitations of the software. Users should be familiar with the many default values on key parameters and default settings and, if possible, how to make changes on these values. Small toy problems should be tried to gain confidence on the capability of the software before serious applications are implemented.

VIII. Model Evaluation and Comparison

Once the modeling process is completed, model performance must be tested or validated using the data not used in the model-building stage. In addition, as ANNs are often used as a nonlinear alternative to traditional statistical models, the performance of ANNs needs to be compared to that of conventional methods to show the value of ANN models. As noted in [1], "if such a comparison is not conducted, it is difficult to argue that the study has taught us much about the value of ANNs."

There are many problems in neural network research with regard to an appropriate evaluation and comparison of neural network models. According to Adya and Collopy [1], "a significant portion of the ANN research in forecasting and prediction lacks validity" in terms of: 1) comparison with established methods; 2) use of true out-of-sample for testing; and 3) use of a reasonable test sample size. Flexer [45] criticizes the lack of statistical evaluations in published studies and proposes the minimum requirements for such evaluations.

As pointed out earlier, one of the major pitfalls in the literature is the failure to use independent holdout samples for out-of-sample evaluations. Sometimes, the holdout sample is clearly not used and other times, it is not clear if the holdout sample is used to find the best model or fine tune the network parameters. It is worthwhile to reemphasize that the holdout (or test) sample should not be used in the model estimation and selection process. If it is, then it is not a genuine out-of-sample. Unfortunately, many ANN researchers overlook this point and regard the accuracy measures obtained in this way as being a true out-of-sample testing result. As the real prediction accuracy will be generally worse than that found for the holdout sample that has already been used in the model-building process, it is almost certain that the reported ANN performance is overstated.

Prechelt [92] examines nearly 200 papers on ANN learning algorithms published in four leading neural network journals and finds that many of them are not evaluated thoroughly enough to be "acceptable." He defines the criteria for acceptable evaluation as: 1) use of at least two real problems; and 2) comparison with at least one alternative algorithm. Unfortunately, 78% of published studies do not meet this minimum standard. In a survey [45] of two leading journals in neural networks, only three out of 43 papers clearly use the third independent data set. Adya and Collopy [1] find that 21 out of 48 (44%) studies in business applications are not effectively evaluated or validated. Vellido *et al.* [119] give similar findings. Maier and Dandy [84] report that among 43 ANN applications to water resource variable forecasting, "only two papers used an independent test set in addition to the training and validation sets." In a review of neural networks used for short-term load forecasting in the last decade, Hippert *et al.* [61] conclude that most ANN models are "not systematically tested." The same problem is also reported by a number of other survey studies such as [137] and [140].

To adequately evaluate the true performance of a model, enough sample size should be allocated to the holdout test sample. With too few observations in the testing set, it is possible that the results obtained are due to chance. One of the problems reported in [1] is the insufficient sample size for validation. For evaluation purposes, they find that 40 or more cases for classification and 75 or more observations for forecasting are reasonable. Ashley [5] has recently studied the issue of how much out-of-sample data are necessary in order for the forecasting improvement to be statistically significant. His simulation results suggest that at least 100 observations are necessary in order for a 20% forecasting error reduction to be statistically significant at the 5% level.

A major problem in evaluating the ANN classifiers is the failure to consider the relative cost of misclassification. Many studies only use the overall classification rate as the sole performance measure of the capability of the model, taking no account of different misclassification errors, which are typically more critical for various decision-making situations. Berardi and Zhang [139] discuss the effect of unequal misclassification costs on classification as well as decision making and find that misclassification cost could have significant impact on the classification results and ignoring the cost information could adversely affect the decision making. In a survey of neural network applications in auditing and risk assessment, Calderon and Cheh [137] find that almost all the studies do not take different costs into consideration.

Many pitfalls lie in the inappropriate comparison of ANN models with other models in statistics, machine learning, and data mining. Either there is no comparison at all or the comparison is not done in an entirely satisfactory manner. Often the comparison is made based on simple measures of accuracy, and statistical significance is mostly overlooked. In addition, the comparison to important benchmark models is not often conducted. For example, in the financial time series literature, the random walk model often emerges as the dominant one among many linear and nonlinear methods. In classification problems, it is well known that by chance alone, one can achieve high-hit rate if the classes are very unbalanced. Therefore, it is important to compare the performance of neural networks to that of the benchmarks. It is possible that ANNs may outperform another statistical model but both fail to show significant improvement over the benchmark models.

Other pitfalls on comparing classifiers have been pointed out by several authors [38], [92], [104]. Duin's [38] major concern is the user-dependent nature of neural network results, and neural network investigators may have "a strong temptation" to do more tuning if they find their models performing relatively poorly. Therefore, the performance reported can be biased. Prechelt [92] reports that 33% of 190 studies he examined have no comparison with other algorithms. Salzberg [104] finds that the literature largely ignores the experimentwise overall error in comparing multiple classifiers and discusses the major problem of the practice of using simple t-tests to compare multiple algorithms on multiple data sets even if the test sets are not independent.

Adya and Collopy [1] propose the following three validation criteria to objectively evaluate the performance of ANNs: 1) comparing to well-accepted models; 2) using true out-of-samples; and 3) ensuring enough sample size in the out-of-sample (40 for classification and 75 for time series forecasting).

Statistical testing should be considered in most of the comparisons. As many comparisons are based on the same holdout sample, special matched sample statistical procedures can be used. To compare accuracies of several classifiers, the F^+ statistic based on repeated measures analysis and described by Looney [82] is recommended. If only two classifiers are involved, then the binomial test [104] for dependent samples or the Goldstein test [53] for independent samples can be used. In addition, the Kappa statistic has been increasingly recommended as an appropriate measure for agreement between classifiers when the data are categorical [149], [150]. For time series forecasting, researchers should consider the Diebold–Mariano test [37].

Many studies use only one training sample and one validation or testing sample to compare different models. Results based on one comparison may suffer from sample biases [106] or random influences [45]. This is largely due to the unstable nature of the neural network model in model building and estimation [13], [122]. Therefore, it is often desirable to use multiple runs or samples. Bootstrap and other resampling techniques are useful in this regard to evaluate ANN performance statistically [25]. Major multiple sample approaches to classifier evolutions can be found in [72]. Although bootstrapping may be difficult for time series forecasting problems, using multiple cross-validation techniques [68], [74] and multiple test periods [115] can be helpful.

It is also important to note that while summary measures such as the overall error rate or average absolute error are useful to give overall performance of the model, they do not provide useful information for decision makers regarding individual case decisions. In addition, they do not provide evidence on the quality of each point estimate or prediction. For classification problems, the receiver-operating characteristics (ROC) curve [57] is better used as it is a more comprehensive performance measure than the single classification error measure. One important feature of the ROC curve is its ready incorporation of prevalence and misclassification cost factors, which are critical for many decision-making problems where different misclassification errors carry significantly uneven consequences. Lisboa [148] points out that although ROC is the *de facto* standard in medical diagnosis, it is unfortunate that "scant attention is given to the usually skewed nature of the data." On the other hand, confidence or prediction intervals should also be used in conjunction with the point estimates or predictions for forecasting especially time series forecasting problems [26], [35], [70], [103]. The use of confidence interval provides a means to assess the reliability of the model and its estimates.

IX. Publication Bias

Publication bias against nonsignificant results can promote some pitfalls in neural network research. It encourages data snooping by repeatedly tuning the model architectures and other parameters if initial results on the holdout sample are not satisfactory. Although mixed findings do appear in published studies, the general tendency in neural network research discourages the negative results reported with ANNs. Overall, it is much easier to publish positive findings than to publish negative results [8], [19].

Many large quantitative competitions in forecasting [86] and classification [88] show that no single method including neural networks is dominantly the best for every problem in every situation. Thus, to prove that neural networks are universally the best can be futile. However, in ANN research, studies are plentiful aiming to show and claim the best performance of ANNs for all problems. Often times, the conclusion is drawn on the limited empirical evidence based on a limited number of data sets. Therefore, readers of these findings must be aware of the fact that the obtained results may not be able to generalize beyond the data sets used in the particular study. It is quite possible that a particular method or algorithm for ANNs works well for a fairly large number of problems, but there are possibly many other problems or data sets, for which the method may perform badly.

The problem can be this. If a researcher tried a number of data sets and chose to report only the good results with his/her method, then the study is easy to get through the review process for publication. However, if he or she honestly reports mixed results with the method, it is difficult or impossible for the study to be published. This is an awkward situation. On one hand, we accept the reality that there is no such thing as the best method or algorithm for any problem. On the other hand, we are not ready to publish mixed results in one study. Of course, with mixed findings, researchers must seek to address under what conditions or for what types of problems the proposed method works best. Addressing such issues, however, is not always an easy task.

Another major problem in the published articles is that many applications of ANNs do not reveal the detail of many aspects of the modeling process including data, data processing, experimental design, model selection, parameter, and other tunings made during the process. This can hide pitfalls and misuses of the technique employed in studies. It will also affect the objective and rigorous evaluation of the methods used. More importantly, it prevents the possibility of replication, a "key requirement of genuine scientific progress" [19].

Reviewers of academic journals should set high standards in reviewing articles. They should demand more detailed descriptions on several key modeling parameters as well as the model-building experiment. Analysis of the results is certainly important but only after the modeling exercise is done flawlessly. On the other hand, we should be more open-minded to accept mixed or negative findings toward ANNs for well-planned and executed studies.

X. Conclusion

In this paper, we present many pitfalls in neural network research and applications. We believe that the key to avoiding pitfalls in neural network research is the awareness of the potential pitfalls and their harms to the research study. It is important to realize that there are numerous ways that ANN techniques can be misapplied and misused. Unwary investigators are more likely to incur pitfalls. Furthermore, an awareness of the problems can lead to healthy skepticism and higher standards in the interpretation of reported findings in the literature. We also offer many insights and good practices that can help neural network researchers and practitioners in their endeavor to improve the quality of research and application.

There is no doubt that ANNs can be one of the most useful tools in one's toolkit for quantitative modeling. ANNs are good alternative candidates to traditional modeling techniques for tasks of pattern recognition, pattern classification, system control, and forecasting. The promises and opportunities of neural network research are evident, judging by the growing literature and numerous exciting business, industrial, and medical applications. Despite the skepticism, the field of neural network is well past the stage of a "passing fad."

However, ANNs are not a panacea for all problems under all environments. They cannot replace all other data analysis methods from statistics, machine learning, and data mining. They are not without problems and difficulties. It is incorrect to believe that like statistics, ANNs are already an established field and the application of ANNs can be as easy as running automated software. It is important to note that the uncritical use of the very flexibility inherent in the nonlinear capabilities of neural networks can easily generate implausible solutions, leading to exaggerated claims of their potentials. It is equally important to note that there are still many practical and theoretical problems that hinder the development and practical use of neural networks.

As pointed out earlier, ANNs can be treated as statistical methods. Consequently, general guidelines and good practices from statistics can and should be considered and followed in ANN research [17], [93]. Neural network community may gain much by listening to Kingman's address to statisticians: "even if statistics can never be a closed profession, it would be both foolish and irresponsible to deny a collective responsibility on the part of the statistical community, a responsibility to ensure the highest possible standards of competence, integrity and judgment. . .." If not, "the credibility of statistics as a whole is threatened" [77].

References

[1] M. Adya and F. Collopy, "How effective are neural networks at forecasting and prediction? A review and evaluation," *J. Forecast.*, vol. 17, pp. 481–495, 1998.

[2] E. C. Almer, *Statistical Tricks and Traps: An Illustrated Guide to the Misuses of Statistics*. Los Angeles, CA: Pyrczak, 2000.

[3] U. Anders and O. Korn, "Model selection in neural networks," *Neural Netw.*, vol. 12, pp. 309–323, 1999.

[4] R. Andrews, J. Diederich, and A. B. Tickle, "Survey and critique of techniques for extracting rules from trained artificial neural networks," *Knowl.-Based Syst.*, vol. 8, no. 6, pp. 373–389, 1995.

[5] R. Ashley, "Statistically significant forecasting improvements: How much out-of sample data is likely necessary?" *Int. J. Forecast.*, vol. 19, pp. 229–239, 2003.

[6] A. R. Bansal, J. Kauffman, and R. R. Weitz, "Comparing the modeling performance of regression and neural networks as data quality varies: A business value approach," *J. Manage. Inf. Syst.*, vol. 10, pp. 11–32, 1993.

[7] A. R. Barron, "A comment on 'Neural networks: A review from a statistical perspective'," *Stat. Sci.*, vol. 9, pp. 33–35, 1994.

[8] C. B. Begg and J. A. Berlin, "Publication bias: A problem in interpreting medical data (with discussion)," *J. R. Stat. Soc. Ser. A*, vol. 151, pp. 419–463, 1988.

[9] M. R. Belli, M. Conti, P. Crippa, and C. Turchetti, "Artificial neural networks as approximators of stochastic processes," *Neural Netw.*, vol. 12, no. 4–5, pp. 647–658, 1999.

[10] J. M. Benitez, J. L. Castro, and I. Requena, "Are artificial neural networks black boxes?" *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 1156–1164, Sep. 1997.

[11] M. Bishop, *Neural Networks for Pattern Recognition*. Oxford: Oxford Univ. Press, 1995.

[12] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. San Francisco, CA: Holden-Day, 1976.

[13] P. Burrascano, M. Battisti, and D. Pirollo, "Finite sample size and neural model uncertainty," *Neurocomputing*, vol. 19, pp. 121–131, 1998.

[14] J. L. Callen, C. C. Y. Kwan, P. C. Y. Yip, and Y. Yuan, "Neural network forecasting of quarterly accounting earnings," *Int. J. Forecast.*, vol. 12, pp. 475–482, 1996.

[15] J. L. Castro, C. J. Mantas, and J. M. Benítez, "Neural networks with a continuous squashing function in the output are universal approximators," *Neural Netw.*, vol. 13, no. 6, pp. 561–563, 2000.

[16] J. L. Castro, I. Requena, and J. M. Benitez, "Interpretation of artificial neural networks by means of fuzzy rules," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 101–116, Jan. 2002.

[17] C. Chatfield, "Avoiding statistical pitfalls," *Stat. Sci.*, vol. 6, no. 3, pp. 240–268, 1991.

[18] ——, "Neural networks: Forecasting breakthrough or just a passing fad?" *Int. J. Forecast.*, vol. 9, pp. 1–3, 1993.

[19] ——, "Positive or negative?" *Int. J. Forecast.*, vol. 11, pp. 501–502, 1995.

[20] ——, "Model uncertainty, data mining and statistical inference," *J. R. Stat. Soc. Ser. A*, vol. 158, pp. 419–466, 1995.

[21] ——, "Model uncertainty and forecast accuracy," *J. Forecast.*, vol. 15, pp. 495–508, 1996.

[22] ——*Time-Series Forecasting*. Boca Raton, FL: Chapman & Hall/CRC, 2001.

[23] T. Chen and H. Chen, "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems," *Neural Netw.*, vol. 6, pp. 911–917, 1995.

[24] B. Cheng and D. Titterington, "Neural networks: A review from a statistical perspective," *Stat. Sci.*, vol. 9, no. 1, pp. 2–54, 1994.

[25] M. R. Chernick, V. K. Murthy, and C. D. Nealy, "Application of bootstrap and other resampling techniques: Evaluation of classifier performance," *Pattern Recognit. Lett.*, vol. 3, pp. 167–178, 1985.

[26] G. Chryssolouris, M. Lee, and A. Ramsey, "Confidence interval prediction for neural network models," *IEEE Trans. Neural Netw.*, vol. 7, no. 1, pp. 229–232, Jan. 1996.

[27] K. B. Church and S. P. Curram, "Forecasting comsumers' expenditure: A comparison between econometric and neural network models," *Int. J. Forecast.*, vol. 12, pp. 255–267, 1996.

[28] A. Ciampi and Y. Lechevallier, "Statistical models as building blocks of neural networks," *Commun. Stat.—Theory Methods*, vol. 26, no. 4, pp. 991–1009, 1997.

[29] J. B. Cohen, "The misuse of statistics," *J. Amer. Stat. Assoc.*, vol. 33, pp. 657–674, 1938.

[30] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Trans. Neural Netw.*, vol. 51, no. 2, pp. 240–254, Mar. 1994.

[31] M. Cottrell, B. Girard, Y. Girard, M. Mangeas, and C. Muller, "Neural modeling for time series: A statistical stepwise method for weight elimination," *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1355–1364, Nov. 1995.

[32] D. R. Cox, "A note on data-splitting for the evaluation of significance levels," *Biometrika*, vol. 62, no. 2, pp. 441–444, 1975.

[33] B. Curry and P. Morgan, "Neural networks: A need for caution," *Omega*, vol. 25, no. 1, pp. 123–133, 1997.

[34] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control Signals Syst.*, vol. 2, pp. 303–314, 1989.

[35] R. D. D. Veaux, J. Schumi, J. Schweinsberg, and L. H. Ungar, "Prediction intervals for neural networks via nonlinear regression," *Technometrics*, vol. 40, no. 4, pp. 273–282, 1998.

[36] F. Denton, "Data mining as an industry," *Rev. Econ. Stat.*, vol. 67, pp. 124–127, 1985.

[37] F. X. Diebold and R. S. Mariano, "Comparing predictive accuracy," *J. Bus. Econ. Stat.*, vol. 13, pp. 253–263, 1995.

[38] R. P. W. Duin, "A note on comparing classifiers," *Pattern Recognit. Lett.*, vol. 17, pp. 529–536, 1996.

[39] R. A. Eisenbeis, "Pitfalls in the application of discriminant analysis in business, finance, and economics," *J. Finance*, vol. 32, no. 3, pp. 875–900, 1977.

[40] D. Touretzky, Ed. Denver, CO:Morgan Kaufinann,S. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Advances in Neural Information Processing Systems*, vol. 2, D. Touretzky, Ed. Denver, CO: Morgan Kaufimann, pp. 524–532.

[41] J. J. Faraway, "On the cost of data analysis," *J. Comput. Graph. Stat.*, vol. 1, pp. 213–229, 1992.

[42] ——, "Data splitting strategies for reducing the effect of model selection on inference," *Comput. Sci. Stat.*, vol. 30, pp. 332–341, 1998.

[43] J. J. Faraway and C. Chatfield, "Time series forecasting with neural networks: A comparative study using the airline data," *Appl. Stat.*, vol. 47, pp. 231–250, 1998.

[44] D. Fletcher and E. Goss, "Forecasting with neural networks—An application using bankruptcy data," *Inf. Manage.*, vol. 24, pp. 159–167, 1993.

[45] A. Flexer, "Statistical evaluation of neural network experiments: Minimum requirements and current practice," in *Proc. 13th Eur. Meeting Cybern. Syst. Res.*, 1996, R. Trappl, Ed., pp. 1005–1008.

[46] I. Flood and N. Kartam, "Neural network in civil engineering—I: Principles and understanding," *J. Comput. Civil Eng.*, vol. 8, no. 2, pp. 131–148, 1994.

[47] M. Frean, "The Upstart algorithm: A method for constructing and training feed-forward networks," *Neural Comput.*, vol. 2, pp. 198–209, 1990.

[48] K. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Netw.*, vol. 2, pp. 183–192, 1989.

[49] P. Gallinari, S. Thiria, R. Badran, and F. Fogelman-Soulie, "On the relationships between discriminant analysis and multilayer perceptrons," *Neural Netw.*, vol. 4, pp. 349–360, 1991.

[50] S. Geman, E. Bienenstock, and T. Doursat, "Neural networks and the bias/variance dilemma," *Neural Comput.*, vol. 5, pp. 1–58, 1992.

[51] R. Gencay and F. Selcuk, "Software reviews: Neural network toolbox 3.0 for use with MATLAB," *Int. J. Forecast.*, vol. 17, pp. 305–322, 2001.

[52] H. Gish, "A probabilistic approach to the understanding and training of neural network classifiers," in *Proc. IEEE Int. Conf. Acoustic, Speech Signal Process.* 3–6, 1990, vol. 3, pp. 1361–1364.

[53] M. Goldstein, "An approximate test for comparative discriminatory power," *Multivariate Behav. Res.*, vol. 11, pp. 157–163, 1976.

[54] W. L. Gorr, "Research prospective on neural network forecasting," *Int. J. Forecast.*, vol. 10, pp. 1–4, 1994.

[55] R. S. Govindaraju and A. R. Rao, "Artificial neural networks in hydrology: A passing fad?" *J. Hydrologic Eng., ASCE*, vol. 5, no. 3, pp. 225–226, 2000.

[56] C. W. J. Granger, "Strategies for modeling nonlinear time-series relationships," *Econ. Rec.*, vol. 69, pp. 233–238, 1993.

[57] D. J. Hand, *Construction and Assessment of Classification Rules*. Chichester, U.K.: Wiley, 1997.

[58] ——, "Data mining: Statistics and more?," *Amer. Stat.*, vol. 52, no. 2, pp. 112–118, 1998.

[59] E. J. Hartman, J. D. Keeler, and J. M. Kowalski, "Layered neural networks with Gaussian hidden units as universal approximations," *Neural Comput.*, vol. 2, pp. 210–215, 1990.

[60] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag, 2001.

[61] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term forecasting: A review and evaluation," *IEEE Trans. Power Syst.*, vol. 16, no. 1, pp. 44–55, Feb. 2001.

[62] R. Hooke, *How to Tell the Liars from the Statisticians*. New York: Marcel Dekker, 1983.

[63] R. G. Hoptroff, "The principles and practice of time series forecasting and business modeling using neural networks," *Neural Comput. Appl.*, vol. 1, pp. 59–66, 1993.

[64] K. Hornik, "Approximation capabilities of multilayer feed-forward networks," *Neural Netw.*, vol. 4, pp. 251–257, 1991.

[65] ——, "Some new results on neural network approximation," *Neural Netw.*, vol. 6, pp. 1069–1072, 1993.

[66] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, pp. 359–366, 1989.

[67] ——, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Netw.*, vol. 3, pp. 551–560, 1990.

[68] M. Y. Hu, G. P. Zhang, C. X. Jiang, and B. E. Patuwo, "A cross-validation analysis of neural network out-of-sample performance in exchange rate forecasting," *Decision Sci.*, vol. 30, pp. 197–216, 1999.

[69] D. Huff, *How to Lie with Statistics*. New York: Norton, 1954.

[70] J. T. G. Hwang and A. A. Ding, "Prediction intervals for artificial neural networks," *J. Amer. Stat. Assoc.*, vol. 92, pp. 748–757, 1997.

[71] O. Intrator and N. Intrator, "Interpreting neural-network results: A simulation study," *Comput. Stat. Data Anal.*, vol. 37, pp. 373–393, 2001.

[72] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[73] B. A. Jain and B. N. Nag, "Performance evaluation of neural network decision models," *J. Manage. Inf. Syst.*, vol. 14, no. 2, pp. 201–216, 1997.

[74] I. Kaastra and M. Boyd, "Designing a neural network for forecasting financial and economic time series," *Neurocomputing*, vol. 10, pp. 215–236, 1996.

[75] G. A. Kimble, *How to Use (and Misuse) Statistics*. Englewood Cliffs, NJ: Prentice-Hall, 1978.

[76] G. King, "How not to lie with statistics: Avoiding common mistakes in quantitative political science," *Amer. J. Pol. Sci.*, vol. 30, pp. 666–687, 1986.

[77] J. Kingman, "Statistical responsibility," *J. R. Stat. Soc. Ser. A*, vol. 152, pt. 3, pp. 277–285, 1989.

[78] B. D. Klein and D. F. Rossin, "Data quality in neural network models: Effect of error rate and magnitude of error on predictive accuracy," *Omega*, vol. 27, pp. 569–582, 1999.

[79] K. A. Kracha and U. Wagner, "Applications of artificial neural networks in management science: A survey," *J. Retailing Consum. Services*, vol. 6, pp. 185–203, 1999.

[80] K. Laudon, "Data quality and due process in large interorganizational record systems," *Commun. ACM*, vol. 29, no. 1, pp. 4–11, Jan. 1986.

[81] B. LeBaron and A. S. Weigend, "A bootstrap evaluation of the effect of data splitting on financial time series," *IEEE Trans. Neural Netw.*, vol. 9, no. 1, pp. 213–220, Jan. 1998.

[82] S. W. Looney, "A statistical technique for comparing the accuracies of several classifiers," *Pattern Recognit. Lett.*, vol. 8, pp. 5–9, 1988.

[83] C. N. Lu, H. T. Wu, and S. Vemuri, "Neural network based short term load forecasting," *IEEE Trans. Power Syst.*, vol. 8, no. 1, pp. 336–342, Feb. 1993.

[84] H. R. Maier and G. C. Dandy, "Neural networks for the prediction and forecasting of water resources variables: A review of modeling issues and applications," *Environ. Model. Softw.*, vol. 15, pp. 101–124, 2000.

[85] S. Makridakis, A. Anderson, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, P. Parzen, and R. Winkler, "The accuracy of extrapolation (time series) methods: Results of a forecasting competition," *J. Forecast.*, vol. 1, pp. 111–153, 1982.

[86] S. Makridakis and M. Hibon, "The M3-Competition: Results, conclusions and implications," *Int. J. Forecast.*, vol. 16, pp. 451–476, 2000.

[87] R. Meese and J. Geweke, "A comparison of autoregressive univariate forecasting procedures for macroeconomic time series," *J. Bus. Econ. Stat.*, vol. 2, pp. 191–200, 1984.

[88] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, *Machine Learning, Neural and Statistical Classification*. New York: Ellis Horwood, 1994.

[89] M. Nelson, T. Hill, T. Remus, and M. O'Connor, "Time series forecasting using NNs: Should the data be deseasonalized first?" *J. Forecast.*, vol. 18, pp. 359–367, 1999.

[90] K. Ord, "Commentaries on the M3-competition: An introduction, some comments and a scorecard," *Int. J. Forecast.*, vol. 17, pp. 537–584, 2001.

[91] Y. R. Park, T. J. Murray, and C. Chen, "Predicting sun spots using a layered perceptron neural network," *IEEE Trans. Neural Netw.*, vol. 7, no. 2, pp. 501–505, Mar. 1996.

[92] L. Prechelt, "A quantitative study of experimental evaluations of neural network learning algorithms: Current research practice," *Neural Netw.*, vol. 9, no. 3, pp. 457–462, 1996.

[93] D. A. Preece, "Good statistical practice," *Statistician*, vol. 36, pp. 397–408, 1987.

[94] M. Qi, "Nonlinear predictability of stock returns using financial and economic variables," *J. Bus. Econ. Stat.*, vol. 17, pp. 419–429, 1999.

[95] M. Qi and G. P. Zhang, "An investigation of model selection criteria for neural network time series forecasting," *Eur. J. Oper. Res.*, vol. 132, pp. 666–680, 2001.

[96] J. Racine, "On the nonlinear predictability of stock returns using financial and economic variables," *J. Bus. Econ. Stat.*, vol. 19, pp. 380–382, 2001.

[97] S. Raudys, "Evolution and generalization of a single neuron—I: Single-layer perceptron as seven statistical classifiers," *Neural Netw.*, vol. 11, pp. 283–296, 1998.

[98] S. J. Raudys and A. K. Jain, "Small sample size effects in statistical pattern recognition: Recommendations for practitioners," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 3, pp. 252–264, Mar. 1991.

[99] R. Reed, "Pruning algorithms—A survey," *IEEE Trans. Neural Netw.*, vol. 4, no. 5, pp. 740–747, Sep. 1993.

[100] M. D. Richard and R. Lippmann, "Neural network classifiers estimate Bayesian *a posteriori* probabilities," *Neural Comput.*, vol. 3, pp. 461–483, 1991.

[101] B. D. Ripley, "Statistical aspects of neural networks," in *Networks and Chaos—Statistical and Probabilistic Aspects*, O. E. Barndorff-Nielsen, J. L. Jensen, and W. S. Kendall, Eds. London, U.K.: Chapman & Hall, 1993, pp. 40–123.

[102] ——, "Neural networks and related methods for classification," *J. R. Stat. Soc. Ser. B*, vol. 56, no. 3, pp. 409–456, 1994.

[103] I. Rivals and L. Personnaz, "Construction of confidence intervals for neural networks based on least squares estimation," *Neural Netw.*, vol. 13, pp. 463–484, 2000.

[104] S. L. Salzberg, "On comparing classifiers: Pitfalls to avoid and a recommended approach," *Data Min. Knowl. Discov.*, vol. 1, pp. 317–328, 1997.

[105] M. Schumacher, R. Robner, and W. Vach, "Neural networks and logistic regression—Part I," *Comput. Stat. Data Anal.*, vol. 21, pp. 661–682, 1996.

[106] R. Sharda, "Neural networks for the MS/OR analyst: An application bibliography," *Interfaces*, vol. 24, no. 2, pp. 116–130, 1994.

[107] R. Sharda and R. B. Patil, "Connectionist approach to time series prediction: An empirical test," *J. Intell. Manuf.*, vol. 3, pp. 317–323, 1992.

[108] M. Smith, *Neural Networks for Statistical Modeling*. New York: Reinhold, 1993.

[109] R. D. Snee, "Validation of regression models: Methods and examples," *Technometrics*, vol. 19, pp. 415–428, 1977.

[110] D. Srinivasan, A. C. Liew, and C. S. Chang, "A neural network short-term load forecaster," *Elec. Power Syst. Res.*, vol. 28, pp. 227–234, 1994.

[111] S. Stern, "Neural networks in applied statistics," *Technometrics*, vol. 38, no. 3, pp. 205–214, 1996.

[112] N. R. Swanson and H. White, "A model-selection approach to assessing the information in the term structure using linear models and artificial neural networks," *J. Bus. Econ. Stat.*, vol. 13, pp. 265–75, 1995.

[113] ——, "A model selection approach to real-time macroeconomic forecasting using linear models and artificial neural networks," *Rev. Econ. Stat.*, vol. 79, pp. 540–550, 1997.

[114] Z. Tang and P. A. Fishwick, "Feedforward neural nets as models for time series forecasting," *ORSA J. Comput.*, vol. 5, no. 4, pp. 374–385, 1993.

[115] L. J. Tashman, "Out-of-sample tests of forecasting accuracy: An analysis and review," *Int. J. Forecast.*, vol. 16, pp. 437–450, 2000.

[116] A. B. Tickle, R. Andrews, M. Golea, and J. Diederich, "The truth will come to light: Directions and challenges in extracting the knowledge embedded within trained artificial neural networks," *IEEE Trans. Neural Netw.*, vol. 9, no. 6, pp. 1057–1068, Nov. 1998.

[117] A. B. Tickle, F. Maire, G. Bologna, R. Andrews, and J. Diederich, "Lessons from past, current issues, and future research directions in extracting the knowledge embedded in artificial neural networks," in *Hybrid Neural Systems*, S. Wermter and R. Sun, Eds. Berlin, Germany: Springer-Verlag, 2000, pp. 226–239.

[118] J. C. van Houwelingen, "Use and abuse of variance models in regression," *Biometrics*, vol. 44, no. 4, pp. 1073–1081, 1988.

[119] A. Vellido, P. J. G. Lisboa, and J. Vaughan, "Neural networks in business: A survey of applications (1992–1998)," *Expert Syst. Appl.*, vol. 17, pp. 51–70, 1999.

[120] S. Walczak, "An empirical analysis of data requirements for financial forecasting with neural networks," *J. Manage. Inf. Syst.*, vol. 17, no. 4, pp. 203–222, 2001.

[121] E. A. Wan, "Neural network classification: A Bayesian interpretation," *IEEE Trans. Neural Netw.*, vol. 1, no. 4, pp. 303–305, Dec. 1990.

[122] S. Wang, "The unpredictability of standard back propagation neural networks in classification applications," *Manage. Sci.*, vol. 41, no. 3, pp. 555–559, 1995.

[123] B. Warner and M. Misra, "Understanding neural networks as statistical tools," *Amer. Stat.*, vol. 50, no. 4, pp. 284–293, 1996.

[124] H. White, "Learning in artificial neural networks: A statistical perspective," *Neural Comput.*, vol. 1, pp. 425–464, 1989.

[125] ——, "Some asymptotic results for learning in single hidden layer feedforward network models," *J. Amer. Stat. Assoc.*, vol. 84, pp. 1003–1013, 1989.

[126] ——, "Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings," *Neural Netw.*, vol. 3, pp. 535–549, 1990.

[127] ——, "A reality check for data snooping," *Econometrica*, vol. 68, no. 5, pp. 1097–1126, 2000.

[128] R. L. Wilson and R. Sharda, "Bankruptcy prediction using neural networks," *Decis. Support Syst.*, vol. 11, pp. 545–557, 1994.

[129] J. Wyatt, "Nervous about artificial neural networks?" *Lancet*, vol. 346, pp. 1175–1177, 1995.

[130] G. P. Zhang, "Neural networks for classification: A survey," *IEEE Trans. Syst., Man, Cybern. C*, vol. 30, no. 4, pp. 451–462, Nov. 2000.

[131] G. P. Zhang, M. Y. Hu, B. E. Patuwo, and D. C. Indro, "Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis," *Eur. J. Oper. Res.*, vol. 116, pp. 16–32, 1999.

[132] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *Int. J. Forecast.*, vol. 14, pp. 35–62, 1998.

[133] ——, "A simulation study of artificial neural networks for nonlinear time-series forecasting," *Comput. Oper. Res.*, vol. 28, pp. 381–396, 2001.

[134] X. Zhang, "Time series analysis and prediction by neural networks," *Optim. Methods Softw.*, vol. 4, pp. 151–170, 1994.

[135] R. Setiono, W. K. Leow, and J. Zurada, "Extraction of rules from artificial neural networks for nonlinear regression," *IEEE Trans. Neural Netw.*, vol. 13, no. 3, pp. 564–577, May 2002.

[136] R. Setiono and J. Y. L. Thong, "An approach to generate rules from neural networks for regression problems," *Eur. J. Oper. Res.*, vol. 155, pp. 239–250, 2004.

[137] T. G. Calderon and J. J. Cheh, "A roadmap for future neural networks research in auditing and risk assessment," *Int. J. Account. Inf. Syst.*, vol. 3, pp. 203–236, 2002.

[138] J. T. Davis, A. P. Massey, and P. E. R. Lovell, "Supporting a complex audit judgment task: An expert network approach," *Eur. J. Oper. Res.*, vol. 103, pp. 305–372, 1997.

[139] V. L. Berardi and G. P. Zhang, "The effect of misclassification costs on neural network classifiers," *Decis. Sci.*, vol. 30, pp. 659–682, 1999.

[140] G. Schwarzer, W. Vach, and M. Schumacher, "On the misuses of artificial neural networks for prognostic and diagnostic classification in oncology," *Stat. Med.*, vol. 19, pp. 541–561, 2000.

[141] S. D. Balkin and J. K. Ord, "Automatic neural network modeling for univariate time series," *Int. J. Forecast.*, vol. 16, pp. 509–515, 2000.

[142] R. R. Picard and K. N. Berk, "Data splitting," *Amer. Stat.*, vol. 44, pp. 140–147, 1990.

[143] L. Zhao, F. Collopy, and M. Kennedy, "The problem of neural networks in business forecasting: An attempt to reproduce the Hill, O'Connor and Remus study," *Sprouts: Working Papers on Information Environments, Systems, and Organizations*, vol. 3, 2004.

[144] T. Hill, M. O'Connor, and W. Remus, "Neural network models for time series forecasting," *Manage. Sci.*, vol. 42, pp. 1082–1092, 1996.

[145] G. P. Zhang and M. Qi, "Neural network forecasting of seasonal and trend time series," *Eur. J. Oper. Res.*, vol. 160, pp. 501–514, 2005.

[146] C. J. Huberty, "Issues in the use and interpretation of discriminant analysis," *Psychol. Bull.*, vol. 95, no. 1, pp. 156–171, 1984.

[147] P. A. Lachebruch, "Some misuses of discriminant analysis," *Methods Inf. Med.*, vol. 16, pp. 255–258, 1977.

[148] P. J. G. Lisboa, "A review of evidence of health benefit from artificial neural networks in medical intervention," *Neural Netw.*, vol. 15, pp. 11–39, 2002.

[149] R. G. Congalton and K. Green, *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices*. New York: Lewis, 1999.

[150] L. G. Protney and M. P. Watkins, *Foundations of Clinical Research: Applications to Practice*. Princeton, NJ: Prentice-Hall, 2000.

[151] B. K. Wong, T. A. Bodnovich, and Y. Selvi, "Neural network applications in business: A review and analysis of the literature (1988–1995)," *Decis. Support Syst.*, vol. 19, pp. 301–320, 1997.

G. Peter Zhang received the B.Sc. and M.Sc. degrees from East China Normal University, Shanghai, China in 1985 and 1987, respectively, and the Ph.D. degree from Kent State University, Kent, OH, in 1998.

He is an Associate Professor of Managerial Sciences at Georgia State University, Atlanta. His research interests include neural networks, forecasting, time-series analysis, and supply chain management. He is the Editor of the book *Neural Networks in Business Forecasting* (IRM Press, 2004), and the author of more than 50 publications in journals, conferences, and book chapters. He currently serves as an Associate Editor of *Neurocomputing* and is on the editorial review board of *Production and Operations Management* journal.

Dr. Zhang received the Best Paper Award from the IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT in 2004.